

# GeMMINi: Prototipado de interfaces de usuario sobre múltiples dispositivos. Una estrategia basada en Líneas de Producto y MDD

Ignacio Mansanet, Joan Fons, Ismael Torres, Vicente Pelechano

## Resumen

Actualmente existe una gran diversidad de plataformas, entendiendo como plataforma a la combinación de características hardware y software, donde la cantidad de dispositivos crece día a día ofreciendo nuevos escenarios de interacción persona-ordenador. Este trabajo presenta GeMMINi, una solución para desarrollar interfaces especializadas para cada tipo de dispositivo, teniendo en cuenta la naturaleza de estos dispositivos. Se propone un enfoque de desarrollo dirigido por modelos, donde, a partir de modelos abstractos de interfaz y descripciones de dispositivos, se obtienen por transformación de modelos, prototipos de interfaces específicas para cada tipo de dispositivo que serán transformadas a implementaciones nativas en la plataforma destino.

## Palabras clave

Interfaces de Usuario Multi-Dispositivo; Desarrollo Dirigido por Modelos; Líneas de Producto Software; Modelos de Características.

## 1 Introducción

En los últimos años se ha podido observar la aparición de nuevos tipos de dispositivos móviles de pequeño y mediano tamaño, ofreciendo mecanismos de interacción novedosos. Por ejemplo, las interfaces multi-táctiles se han consolidado como una solución más natural en el ámbito móvil, impulsando la penetración de estos dispositivos. Estos dispositivos móviles se suman a los ya existentes (PCs, portátiles...), y la redefinición de otros (como la TV), para ofrecer mayor interactividad. Dada esta situación, podemos asumir que en la actualidad estamos frente a una gran diversidad de dispositivos con diferentes capacidades y características que nos ofrecen múltiples mecanismos para la interacción persona-ordenador. Esta diversidad, también conocida como “fragmentación de dispositivos”, presenta un reto a la comunidad de Interacción Persona-Ordenador (IPO), que requiere soluciones para construir y mantener versiones de aplicaciones para todos los dispositivos, manteniendo la consistencia entre versiones y el tipo de interacción [1].

Existen propuestas de métodos de desarrollo y generación de aplicaciones e interfaces multi-dispositivo, en las que el resultado final de esta generación se realiza sobre un lenguaje común a todos (por ejemplo HTML o XML) [5]. Esta solución es muy versátil ya que la adaptación la realiza el propio dispositivo, pero, por contra, no permite explotar las características (hardware y de interacción) propias de éste (i.e.: GPS, cámara de fotos, pantallas multi-táctiles, etc.). Estas características, no obstante, pueden introducir mejoras en la usabilidad y los mecanismos de interacción de la aplicación por lo que es recomendable contemplarlas en los procesos de desarrollo. Otras propuestas [2,3] abordan la problemática de la fragmentación de dispositivos atendiendo a criterios de modalidad (i.e: el tamaño de la pantalla) pero no a otros como la interacción o la situación del usuario respecto al dispositivo, los cuales son importantes dado que influyen en la manera en que el usuario interactúa con la interfaz (i.e: teclado físico o virtual).

La aportación de este trabajo es GeMMINi, un método para la obtención de múltiples interfaces de usuario adaptadas a la naturaleza de cada dispositivo. GeMMINi utiliza técnicas del Desarrollo Dirigido por Modelos combinadas con conceptos y técnicas de Líneas de Productos Software que permiten (1) describir de manera abstracta los requisitos de interfaz, (2) describir las variantes de los dispositivos, y (3) transformar modelos y generar código para obtener las implementaciones nativas en los dispositivos. Por último, define un catálogo de patrones que permite representar las correspondencias entre conceptos abstractos de interfaz sobre especificaciones concretas para cada tipo de dispositivo.

## 2 GeMMINi: Desarrollando Interfaces de Usuario en Múltiples Dispositivos

Este trabajo presenta GeMMINi, una solución al desarrollo de interfaces de usuario en múltiples dispositivos que utiliza técnicas del DSDM y conceptos y técnicas de LPS. La Figura 1 muestra los componentes (modelos y transformaciones) y las relaciones entre éstos, que se describen a continuación:

### 1. Especificación abstracta de la Interfaz de Usuario.

Descripciones conceptuales (modelos) de interfaces de usuario, que abstraen de problemas relacionados con las características de los dispositivos o plataformas destino y se centran en describir la interfaz del usuario. MOSKitt UIM es el subproyecto dentro de MOSKitt <sup>1</sup> que permite el modelado abstracto de interfaces de usuario. Los constructores abstractos de UIM definen Unidades las que se especifica la Interacción con el usuario, y complementa las descripciones de estructuras de datos (Diagramas de Clases UML), en que se basa para definir la información.

### 2. Especificación de las características de los dispositivos.

Descripción de propiedades y variantes de dispositivos mediante modelos de características, centrada en describir aspectos que

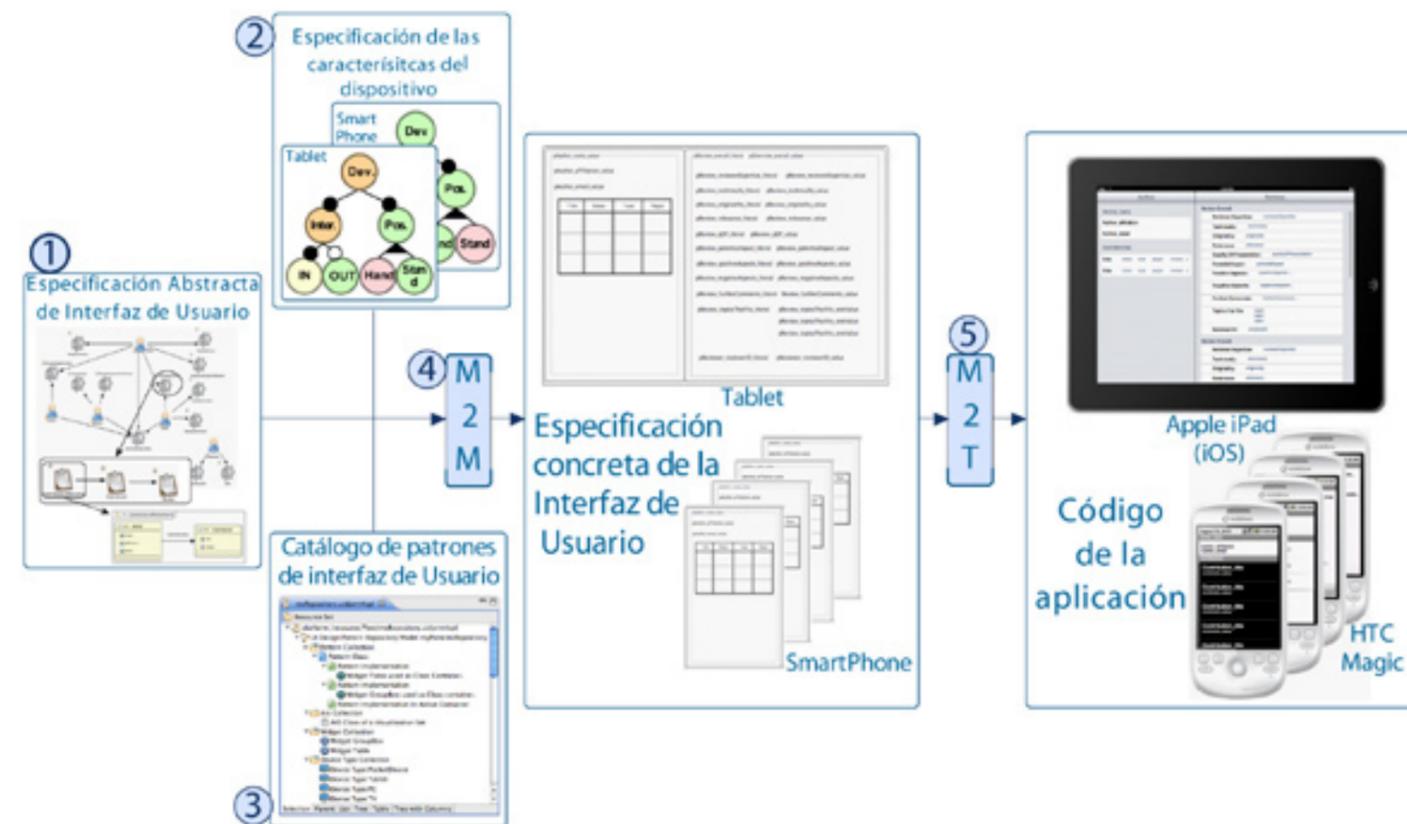


Figura 1. GeMMINi. Componentes del método y las relaciones entre éstos.

afecten a la interacción con los usuarios. Estos modelos son ampliamente utilizados para describir el conjunto de productos derivados de una LPS en términos de características [4]. Nos basamos en una ontología propuesta en [2] que especifica algunos aspectos de interacción, como la entrada por voz o ratón o la salida por proyector o pantalla. Se propone una extensión que refina los modos de entrada y salida, e incorpora otras como el manejo del dispositivo (1 ó 2 manos) o su posición (cerca o lejos del usuario).

**3. Catálogo de patrones de interfaz.** Catálogo de soluciones específicas para cada dispositivo a partir de conceptos abstractos de interfaz. Permite configurar la transformación que obtiene los bocetos. Se diferencian 4 tipos de entidades: los controles para construir los bocetos, los objetos abstractos de interfaz (AIO), los dispositivos, y los patrones que proponen una solución (en forma de widgets) para cada AIO, aportando diferentes implementaciones en función del dispositivo.

<sup>1</sup> <http://www.moskitt.org>

**4. Transformación M2M para obtener bocetos.** Genera modelos de bocetos de interfaz de usuario específicos para cada dispositivo. La transformación analiza el modelo UIM en busca de los constructores abstractos. Para cada uno busca las implementaciones en el catálogo que sean adecuadas para el dispositivo. En caso que exista más de una implementación válida, se consultará al analista, para que elija una implementación. El resultado será un modelo Sketcher. MOSKitt Sketcher es otra herramienta del entorno MOSKitt desarrollada para introducir técnicas para la creación de bocetos en el desarrollo de interfaces de usuario.

**5. Transformación M2T que genera el código de la interfaz.** Generación de la interfaz de usuario a partir de los bocetos y la plataforma destino.

En [6] se puede consultar la aplicación de GeMMINi a un caso práctico: un Sistema de Revisión de Conferencias. Este sistema da soporte al proceso de envío, revisión y notificación de resultados para artículos de investigación en conferencias.

## 3 Conclusiones y Trabajos Futuros

La gran cantidad y diversidad de dispositivos existentes el mercado presentan un amplio abanico de oportunidades de negocio al tiempo que un reto para los desarrolladores de aplicaciones. Existen propuestas que proveen mecanismos para obtener interfaces multi-dispositivo basadas en el uso de un lenguaje común a todos los dispositivos. Pero su naturaleza genérica causa que no se aprovechen las características específicas de cada dispositivo en particular, ya que existen muchos de ellos que no permiten el acceso a ciertas características si no es utilizando los mecanismos que provee el propio sistema operativo presente en el dispositivo.

GeMMINi propone una aproximación que permite el desarrollo de interfaces para múltiples dispositivos partiendo de una especificación abstracta común, abordando el problema de obtener una especificación concreta para cada dispositivo en base a sus características. Los modelos de características nos permiten especificar las capacidades de cada dispositivo, en base a criterios relacionados con la interacción. El uso de los modelos abstractos de interfaz junto con un catálogo de patrones de diseño, permite ofrecer diversas opciones de implementación para un mismo AIO.

Los trabajos futuros son: (1) completar la propuesta definiendo la transformación para la generación de implementaciones nativas, (2) refinarla introduciendo mayor expresividad sobre características de dispositivos y (3) validar la propuesta desarrollando nuevos casos de estudio en otros tipos de dispositivos.

## Agradecimientos

Este trabajo ha sido desarrollado con el soporte del MICINN en el marco del proyecto EVERYWARE TIN2010-18011 y cofinanciado con fondos FEDER.

## Referencias

- 1. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.:** A unifying reference framework for multi-target user interfaces. *Interacting with Computers* 15, pp. 289–308 (2003)
- 2. Clerckx, T., Vandervelpen, C., Coninx, K.:** Engineering Interactive Systems. chap. Task-Based Design and Runtime Support for Multimodal User Interface Distribution, pp. 89–105. Springer-Verlag, Berlin, Heidelberg (2008)
- 3. Gajos, K., Weld, D., Wobbrock, J.:** Automatically generating personalized user interfaces with SUPPLE. In: *Artificial Intelligence vol: 174* 2010, pp. 910–950. Springer Berlin (2010)
- 4. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.:** Feature oriented Domain Analysis (FODA) Feasibility Study. Tech. rep., Carnegie-Mellon University Software Engineering Institute (1990)
- 5. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V.:** USIXML: A Language Supporting Multi-path Development of User Interfaces. In: *Proc. of EHCI '05, LNCS*, vol. 3425, pp. 200–220. Springer Berlin (2005)
- 6. Mansanet, I., Fons, J., Torres, I., Pelechano, V.:** Aplicación de GeMMINI a un caso práctico: Sistema de Revisión de Conferencias. Tech. rep. (2011), <http://www.pros.upv.es/technicalreports/ProS-TR-2011-08.pdf>

## Sobre los autores

**Ignacio Mansanet** investigador miembro del Centro de Investigación en Métodos de Producción de Software de la Universidad Politécnica de Valencia. Sus intereses en investigación incluyen el Desarrollo de Software Dirigido por Modelos, la Interacción Persona-Ordenador y la Computación Móvil y Ubicua. Ignacio Mansanet es Ingeniero Informático y Master en Ingeniería del Software por la Universidad Politécnica de Valencia. Ha publicado artículos en diversas conferencias internacionales. Participa en diferentes proyectos de I+D europeos y con empresas. Contactar con él a través de: [imansanet@pros.upv.es](mailto:imansanet@pros.upv.es)

**Ismael Torres** compagina su trabajo de Investigador con el de Profesor asociado, en el Centro de Investigación en Métodos de Producción de Software de la Universidad Politécnica de Valencia y actualmente, participa en varios proyectos de I+D europeos (OSAMI, Lifewear, Semosa, ...). Sus intereses en investigación se centran en el Desarrollo Dirigido por Modelos, la Inteligencia Ambiental y las Redes Sociales. Ismael Torres es Ingeniero Informático por la Universidad Politécnica de Valencia desde 2000 y Master en Dirección de Empresas por el ESIC en 2007. Ha publicado artículos en revistas científicas y en diversas conferencias internacionales. Email de contacto: [itorres@pros.upv.es](mailto:itorres@pros.upv.es)

**Joan Fons** Profesor Contratado Doctor en el Centro de Investigación en Métodos de Producción de Software de la Universidad Politécnica de Valencia. Obtuvo el doctorado en esta universidad en el 2008 en el ámbito de la Ingeniería Web. Actualmente trabaja en el Desarrollo de Software Dirigido por Modelos, el Desarrollo de Interfaces de Usuario, las Líneas de Producto Software, la Inteligencia Ambiental, la Computación Autónoma y la Evolución del Software. Participa en diferentes proyectos de I+D europeos y con empresas, entre los que cabe destacar su participación en el desarrollo de herramientas industriales para la aplicación del DSDM en MOSKitt ([www.moskitt.org](http://www.moskitt.org)). Ha publicado en revistas y conferencias internacionales de primer nivel. Contactar con él a través de: [jffons@pros.upv.es](mailto:jffons@pros.upv.es).

**Vicente Pelechano** es Profesor Titular de Universidad en el Centro de Investigación en Métodos de Producción de Software de la Universidad Politécnica de Valencia y lidera la supervisión técnica de MOSKitt, una herramienta CASE de código abierto ([www.moskitt.org](http://www.moskitt.org)). Sus intereses en investigación incluyen el Desarrollo de Software Dirigido por Modelos, la Ingeniería Web, la Computación Móvil y Ubicua, y el Modelado de Procesos de Negocio. Pelechano es doctor por la Universidad Politécnica de Valencia desde 2001, ha publicado artículos en revistas científicas de prestigio y en diversas conferencias internacionales. Ha sido miembro de Comités Científicos y de Organización de reconocidas Conferencias Internacionales y Workshops. Contactar con él a través de: [pele@pros.upv.es](mailto:pele@pros.upv.es).

# GeMMINi: Prototipado de interfaces de usuario sobre múltiples dispositivos. Una estrategia basada en Líneas de Producto y MDD

*Actualmente existe una gran diversidad de plataformas, entendiendo como plataforma a la combinación de características hardware y software, donde la cantidad de dispositivos crece día a día ofreciendo nuevos escenarios de interacción persona-ordenador.*

*Este trabajo presenta GeMMINi, una solución para desarrollar interfaces especializadas para cada tipo de dispositivo, teniendo en cuenta la naturaleza de estos dispositivos. Se propone un enfoque de desarrollo dirigido por modelos, donde, a partir de modelos abstractos de interfaz y descripciones de dispositivos, se obtienen por transformación de modelos, prototipos de interfaces específicas para cada tipo de dispositivo que serán transformadas a implementaciones nativas en la plataforma destino.*



© Gregorio G. Reche